DNN-Assisted Particle-Based Bayesian Joint Synchronization and Localization

Meysam Goodarzi, Vladica Sark, Nebojsa Maletic, *Member, IEEE*, Jesús Gutiérrez Terán, *Member, IEEE*, Giuseppe Caire, *Fellow, IEEE*, and Eckhard Grass

***Abstract*— In this work, we propose a Deep neural network-assisted Particle Filter-based (DePF) approach to address the Mobile User (MU) joint synchronization and localization (sync&loc) problem in ultra-dense networks. In particular, DePF deploys an asymmetric time-stamp exchange mechanism between the MUs and the Access Points (APs), which, traditionally, provides us with information about the MUs' clock offset and skew. However, information about the distance between an AP and an MU is also intrinsic to the propagation delay experienced by the exchanged time-stamps. In addition, to estimate the angle of arrival of the received synchronization packets, DePF draws on the multiple signal classification algorithm that is fed with the Channel Impulse Response (CIR) experienced by the sync packets. The CIR is also leveraged to determine the link condition, i.e. Line-of-Sight (LoS) or Non-LoS. Finally, to perform joint sync&loc, DePF capitalizes on particle Gaussian mixtures that allow for a hybrid particle-based and parametric Bayesian Recursive Filtering (BRF) fusion of the aforementioned pieces of information and, thus, jointly estimates the position and clock parameters of the MUs. The simulation results verify the superiority of the proposed algorithm over the state-of-the-art schemes, especially that of the extended Kalman filter- and linearized BRF-based joint sync&loc. In particular, only drawing on the synchronization time-stamp exchange and CIRs from a single AP, for 90% of the cases, the absolute position and clock offset estimation error remain below 1 meter and 2 nanoseconds, respectively.**

***Index Terms*— 5G, joint synchronization and localization, Bayesian particle Gaussian mixture filter, deep neural network, time-stamp exchange.**

Manuscript received 9 February 2022; revised 11 May 2022; accepted 24 May 2022. Date of publication 3 June 2022; date of current version 15 July 2022. The research leading to these results has received funding from the European Union's Framework Programme Horizon 2020 for research, technological development and demonstration under grant agreement No. 871428 (5G-CLARITY). The associate editor coordinating the review of this article and approving it for publication was Z. Qin. *(Corresponding author: Meysam Goodarzi.)*

Meysam Goodarzi and Eckhard Grass are with IHP—Leibniz-Institut für Innovative Mikroelektronik, 15236 Frankfurt (Oder), Germany, and also with the Department of Computer Science, Humboldt University of Berlin, 10117 Berlin, Germany (e-mail: goodarzi@ihp-microelectronics.com; grass@ihp-microelectronics.com).

Vladica Sark, Nebojsa Maletic, and Jesús Gutiérrez Terán are with IHP—Leibniz-Institut für Innovative Mikroelektronik, 15236 Frankfurt (Oder), Germany (e-mail: sark@ihp-microelectronics.com; maletic@ihp-microelectronics.com; teran@ihp-microelectronics.com).

Giuseppe Caire is with the Chair of Communications and Information Theory, Technical University of Berlin, 10623 Berlin, Germany (e-mail: caire@tu-berlin.de).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCOMM.2022.3180069.

Digital Object Identifier 10.1109/TCOMM.2022.3180069

I. Introduction

THE fifth-generation (5G) of mobile communication networks is expected to deploy Access Points (APs) with a high spatial density to meet the increasing demand for mobile data traffic. As a result, Mobile Users (MUs) are expected to be most of the time in Line-of-Sight (LoS) of several APs. This also lays the ground for an accurate MU localization, which is in particular of crucial importance for services such as user tracking and location-assisted beamforming [1], [2]. Furthermore, such APs are likely to be equipped with antenna arrays and they are expected to support Fine Time Measurement (FTM) capability introduced in several standards, e.g., IEEE 802.11 [3]. The former facilitates the Angle of Arrival (AoA) estimation, while the latter allows for the AP-MU time-stamp exchange, by means of which synchronization and distance measurements are enabled. The synchronization itself also plays a decisive role when performing time-based localization. In particular, for many of the state-of-the-art MU localization techniques to function, the clock parameters of the MUs need to be known (or to be continuously tracked). Therefore, it appears that the MU's clock parameter estimation and MU localization are closely intertwined and need to be tackled jointly.

The joint MU synchronization and localization (sync&loc) problem has been extensively addressed in the literature [4]–[9]. The authors in [5] rely on symmetric inter-agent (AP-MU, inter-MU, and inter-AP) time-stamp exchange and Belief Propagation (BP) to jointly estimate MUs' locations and clock offsets. A similar approach has been adopted by [6], [7] with the aid of asymmetric time-stamp exchange mechanism proposed in [10]. While time-stamp exchange is expected to be supported in 5G networks [3], the high number of message-passings required by BP renders the approach limited in practice. Additionally, [3], [5]–[7], [10] provide the estimation of the sync&loc parameters at MUs, whereas for the location-based services to be delivered, these parameters need to be computed at the network side. Another drawback is the strong assumption of a fully cooperative network (also made in [11]). That is, in addition to inter-AP and AP-MU communications, the MUs can also communicate with each other, which is not envisioned in 5G mobile networks. Nevertheless, the cooperation capability between the APs and the Base Stations (BSs) can be drawn on to perform hybrid synchronization as done in [12], [13], laying the ground for an accurate MU joint sync&loc.

Moreover, in [8], [9], the authors leverage Extended Kalman Filtering (EKF) to obtain the estimation of clock and position parameters in ultra-dense networks. In particular, they assume accurate inter-AP synchronization and perform MU joint sync&loc in the presence of uncertainty about the time of arrival and AoA parameters. The level of uncertainty is then determined based on the derived Cramer-Rao bound. Another approach called Linearized BRF (L-BRF), which is based on linearizing the filter, has been employed in [14], [15], albeit the perfect inter-AP synchronization assumption is lifted. Instead, the APs and their backhauling BSs are assumed to be synchronized using cooperative hybrid synchronization introduced in [12], [13]. While EKF and L-BRF can partially mitigate the destructive impact of nonlinearities in the measurements, they are likely to diverge if a reliable estimate of the initial state is not available [16]. Another weakness of these filters is the underestimation of the covariance matrix. A promising approach, on the one hand, to avoid such shortcomings of EKF/L-BRF, and, on the other hand, to boost the accuracy of position estimation, is estimating the (prediction, measurement likelihood, and posterior) distributions by means of Particle Gaussian Mixture (PGM) filters introduced in [17]. Specifically, in this approach, instead of approximating each distribution as a single Gaussian function, they are approximated with a sum weighted of Gaussian functions, or, alternatively, Gaussian mixtures [18]. Nevertheless, the problem that immediately arises when using PGM filters is dimensionality, rendering the approach computationally expensive for multivariable estimations. To overcome this drawback, we resort to a hybrid parametric and particle-based approach where we capitalize on the linear relations between the measurements and the clock parameters to reduce the dimensionality. In comparison to the standard PF, this approach features a strictly lower estimation variance as a result of Rao-Blackwell's lemma discussed in [19], and leads to more accurate estimates given the same number of particles [18]. Specifically, PGM's performance stands out when the uncertainty increases.

Even the PGM-based localization techniques can suffer from divergence under certain conditions, e.g., improper tuning of the filter's hyper-parameters and faulty measurements, resulting mostly from Non-Line-of-Sight (NLoS) links [20]. The former must be addressed when designing the filter, while the latter can be dealt with using NLoS mitigation methods such as those proposed in [21]–[23]. The technique in [21] relies on the multipath components of NLoS links to enhance the positioning accuracy. However, such a method functions well only in the presence of strong multipath components and prior statistics on NLoS-induced errors. The latter are also estimated and utilized along with trajectory tracking in [22] to perform indoor positioning. The authors in [23], however, take another approach and model the measurement noise by a two-mode mixture distribution and approximate the maximum likelihood estimator using expectation maximization. Such approaches add an extra computation overhead that may not be necessary in dense networks where the LoS probability is around 0.8 and increases with the AP density [24]. Therefore, to mitigate the estimation inaccuracy stemming from the faulty measurements, we draw on NLoS identification techniques to identify the NLoS links and discard them. Such an approach boosts the accuracy and features less complexity compared to the methods proposed in [21]–[23].

There is a wide spectrum of NLoS identification approaches adopted in the literature, e.g., hypothesis testing as in [25], the statistical approach taken in [26], and Machine Learning (ML)-based methods such as that of [27]. However, recently ML algorithms, particularly DNN-based approaches such as AmpN [28], have drawn substantial attention in classification problems. In particular, DNNs exhibit a remarkable performance due to their ability, on one hand, in implementing almost any classifier function, and, on the other hand, in extracting task-related features from the input data [29], [30]. Other approaches such as Support Vector Machine (SVM), or Bayesian sequential testing require human intervention that may be, given the limited intuition, flawed, and erroneous. Furthermore, DNN units are also expected to be part of the communication devices as they are the cornerstone of many solutions for different communication problems such as slice management and anomaly detection [31]. Therefore, a DNN-based NLoS identifier appears to be a reasonable choice. The input to the DNN can be signals containing class-relevant features such as received signal strength or Channel Impulse Response (CIR). The CIR turns out to be more informative about the communication environment and link condition. Therefore, for the sake of prediction accuracy, we rely on AP-MU CIRs in this work.

In addition to NLoS-identification, the CIR can also be fed into one of the state-of-the-art AoA estimation algorithms to obtain the signal's direction of arrival. AoA estimation has been extensively investigated in the literature. Algorithms such as MUSIC [32], reduced-dimension MUSIC [33], and ESPRIT [34] can accurately estimate the AoA. A detailed comparison between them has been conducted in [35] concluding that the difference is negligible, albeit MUSIC slightly outperforms the others and, therefore, it is employed for the purpose of this work.

In this paper, we propose a DNN-assisted PF-based (DePF) joint sync&loc algorithm that draws on the CIR to estimate the AoA and to determine the link condition, i.e., LoS or NLoS, thereby excluding the faulty measurements to enable a more precise parameter estimation. It then estimates the joint probability distribution of MU's clock and position parameters using a PGM filter. The dimension of the PGM filter is then reduced by revealing and exploiting the existing linear sub-structures in the measurements, thereby tackling the dimensionality problem. To the best of our knowledge, this is the first work employing a PGM filter in a hybrid particle-based and parametric manner to perform joint sync&loc.

The contribution of this paper is summarized as follows:

- We present and discuss the principles of asymmetric time-stamp exchange and AoA estimation. The former assists in the estimation of the clock skew, offset, and the AP-MU distance, while the latter aids in the position estimation by providing the direction of an MU relative to the position of its serving APs.
- We develop a DNN for NLoS identification based on AP-MU CIRs. The outcome of such a DNN helps to

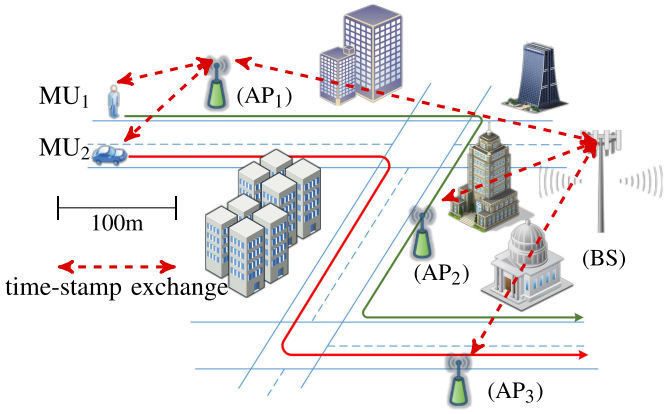


Fig. 1. An example where MU joint sync&loc can be carried out.

identify erroneous measurements, i.e., time-stamps and AoAs, and discard them, thereby preventing large errors in the estimation.

- We propose a DNN-assisted particle filter-based joint sync&loc algorithm that estimates the clock parameters and the position of an MU in a hybrid parametric and particle-based manner. Such an approach not only boosts the estimation accuracy but also overcomes the dimensionality problem that arises in particle Gaussian mixture filters due to the high number of parameters.
- We analyze the performance of the proposed approach with the aid of detailed simulations in a challenging real-world scenario. In particular, the MUs' movement profile comprises acceleration, deceleration, and constant speed. Furthermore, the APs are distributed to provide signal coverage for the MUs.

The rest of this paper is structured as follows: In Section II, we introduce the system model and the preliminaries. Section III describes the details of the DePF algorithm for joint estimation of the clock and position parameters. Furthermore, the simulation results are presented and discussed in Section IV. Finally, Section V concludes this work and points to future works.

Notation: The boldface capital $\boldsymbol{A}$ and lower case $\boldsymbol{a}$ letters denote matrices and vectors, respectively. The n-th element of vector $\boldsymbol{a}$ is indicated by $\boldsymbol{a}[n]$. The symbol "$\cdot$" shows the inner scalar product of two (or multiple) vectors of the same dimension. Moreover, $\boldsymbol{I}_N$ and $\mathbf{0}_N$ represents $N \times N$ dimensional identity and all-zero matrices, respectively. $\mathbf{1}_N$ indicates an N-element all-one vector. Notation $\mathcal{U}(a, b)$ denotes a continuous uniform probability distribution in the interval between a and b with the probability level of $\frac{1}{b-a}$. Furthermore, $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ stands for *probability density function* (pdf) of a Gaussian random vector $\mathbf{x}$ with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. A diagonal matrix with the diagonal elements $(x_1, \cdots, x_K)$ is denoted by $\text{diag}(x_1, \cdots, x_K)$. Symbol $\sim$ stands for "is distributed as" and the symbol $\propto$ represents the linear scalar relationship between two real-valued functions.

II. System Model and Preliminaries

We consider a network of multiple APs with known locations, all backhauled by BSs. The APs are assumed to feature multiple-input multiple-output Uniform Planar Arrays (UPAs), which allow for accurate azimuth and elevation AoA estimations. A further assumption is that they are able to continuously synchronize themselves with the backhauling BSs using the hybrid synchronization algorithm described in [12], [15]. This, in particular, guarantees a low time error among the neighboring APs, enabling a more precise cooperative localization. Moreover, at each sync&loc period T, a set of APs denoted by $\mathcal{I}_i$, can periodically exchange time-stamps with the i-th MU using the FTM feature embedded in the communication devices and implemented by an existing protocol, e.g., precision time protocol [36]. From the packet containing these time-stamps, the APs can also estimate the CIRs and AoA. The AP-MU link condition is probabilistically determined and can be either LoS or NLoS. It is known from [24], that, for such a scenario, the LoS probability is around 0.8, even growing to 0.95 when the AP density is 40 meters. A DNN trained using CIRs is employed to distinguish the LoS condition from NLoS, permitting the localization unit to neglect the measurements conducted under the NLoS condition, thereby augmenting the accuracy of synchronization and localization. In what follows, we firstly present the clock model for the APs and the MUs. Then, we explain the time-stamp exchange mechanism in detail. Subsequently, we discuss the DNN that allows for a reliable NLoS identification. Lastly, the principles of MUSIC algorithm are briefly described.

A. Clock Model

We begin with defining a clock model for MUs and APs. For each node i, we can write

$$c_i(t) = \gamma_i t + \theta_i, \tag{1}$$

where t represents the global reference time. Furthermore, γ_i and θ_i denote the clock skew and the clock offset, respectively. Although the parameter γ_i is generally random and time-varying, it is common to assume that it remains constant in the course of one synchronization period T [37]–[39]. Given that, the first goal of the joint sync&loc algorithm is to estimate and track the clock parameters γ_i and θ_i (or transformations thereof) for each MU. In the sequel, we further clarify the components constructing θ_i as well as the time-stamp exchange mechanism required to estimate the above-mentioned parameters.

B. Offset Decomposition and Time-Stamp Exchange

1) Offset Decomposition: To elaborate on the constituents of the offset θ_i, we break it down as shown in Figure 2. The parameter t_j/t_i is the time taken for a packet to leave the transmitter after being time-stamped, d_{ji}/d_{ij} represents the distance between the nodes j/i and i/j, v_c is the speed of light, and r_i/r_j represents the time that a packet needs to reach the time-stamping point upon arrival at the receiver. Generally, the packets sent from node j to node i do not necessarily experience the same delay as those sent from node i to node j. In other words,

$$t_j + \frac{d_{ji}}{v_c} + r_i \neq t_i + \frac{d_{ij}}{v_c} + r_j.$$

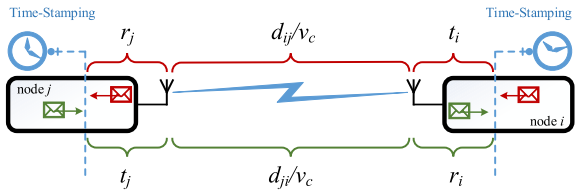


Fig. 2. Decomposition of the clock offset into its constituent components.

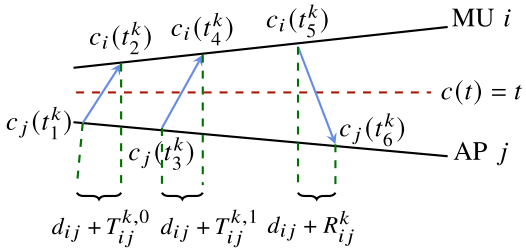


Fig. 3. Asymmetric time-stamp exchange between MU i and AP j.

The variables $T_{ij} = t_j + r_i$, and $R_{ij} = t_i + r_j$ (and correspondingly t_j, t_i, r_j, and r_i,) are random variables due to multiple hardware-related random independent processes and can, therefore, be assumed i.i.d. Gaussian random variables, whereas d_{ji} and d_{ij} are usually assumed to be deterministic and symmetric ($d_{ji} = d_{ij}$) [1], [38]. The random variables T_{ij} and R_{ij} are assumed to be distributed as $\mathcal{N}(T_{ij}|\mu_T, \sigma_T^2)$ and $\mathcal{N}(R_{ij}|\mu_R, \sigma_R^2)$, respectively. As mentioned in [37], [38], [40], while it is typical to assume that $\mu_T = \mu_R$, and parameters σ_T and σ_R are known, having any information about the value of μ_T and μ_R is highly unlikely. Therefore, we construct the joint sync&loc algorithm assuming no knowledge on μ_T and μ_R except for $\mu_T = \mu_R$.

2) Time-Stamp Exchange Mechanism: We draw on the asymmetric time-stamp exchange mechanism shown in Figure 3, proposed in [10], and employed in [6], [37]. Node j transmits a *sync* message wherein the local time $c_j(t_1^k)$ is incorporated. Node i receives the packet and records the local reception time $c_i(t_2^k)$. After a certain time, the process repeats again with $c_j(t_3^k)$ and $c_i(t_4^k)$. Subsequently, at local time $c_i(t_5^k)$, node i sends back a *sync* message to node j with $c_i(t_2^k)$, $c_i(t_4^k)$ and $c_i(t_5^k)$ incorporated. Upon reception, node j records the local time $c_j(t_6^k)$. Given this mechanism, at the k-th round of time-stamp exchange (and correspondingly k-th round of joint sync&loc), we expect the localization unit to have collected the time-stamps

$$\mathbf{c}_{ij}^k = \left[c_j(t_1^k), c_i(t_2^k), c_j(t_3^k), c_i(t_4^k), c_i(t_5^k), c_j(t_6^k)\right].$$

The collected time-stamps will be exploited in Section III to design a joint sync&loc algorithm. In the following subsections, we firstly use the CIRs to identify whether the MU-AP link condition is LoS or NLoS. Later on, the same CIRs are utilized to estimate the AoA.

C. NLoS Identification and Channel Impulse Response

The ability to estimate the CIR is highly ubiquitous among the APs. Therefore, relying on the CIR to develop a localization algorithm appears to be a realistic approach. The AP-MU

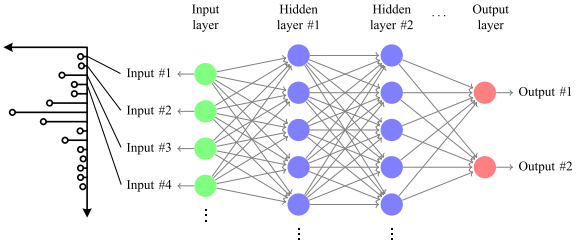


Fig. 4. The DNN employed for NLoS-identification. It has $l_H = 2$ hidden layers with n_H neurons and two output neurons.

CIR is a rich source of information about the condition of the communication link, e.g., LoS or NLoS, and the location of the MU. More precisely, the former is crucial to know when estimating the latter as the accuracy of the distance/time and AoA measurements significantly decline when conducted under NLoS conditions.

Figure 4 shows the architecture of the DNN deployed for NLoS-identification. The input layer has one channel fed with N samples, i.e., the magnitude of the CIR. The number of hidden layers and neurons in each hidden layer is set to l_H and n_H, respectively. The rationale to rely on when selecting these numbers is that, according to [29], any classifier function can be realized by two hidden layers, i.e., currently there is no theoretical reason to use more than two. However, the lack of evidence does not imply that the DNNs with more hidden layers do not improve the accuracy of classification, it rather suggests that the number of required hidden layers does not follow a well-established logic and is mostly determined by a trial-and-error process. Therefore, for the algorithm proposed in this work, we empirically determine the l_H that delivers the best performance. Furthermore, as a rule of thumb, the number of neurons is suggested to be between the number of inputs and that of the outputs to prevent under/overfitting.

Let the output probability vector of the DNN be $[1 - \hat{p}_{\text{nlos}}, \hat{p}_{\text{nlos}}]$, where $\hat{p}_{\text{nlos}}$ denotes the probability of the CIR being corresponded to an NLoS link. For the NLoS-identifier, we seek to train the DNN such that the output probability vector is as close as possible to $[1, 0]/[0, 1]$ for the LoS/NLoS CIRs. In other words, from the optimization point of view, we aim to design a loss function whose output is small when the DNN returns the correct vector and it is large otherwise. It turns out that the function that possesses the above-mentioned property is the logarithmic function [41]. Mathematically, the loss function is given by [30]

$$\mathcal{L} = -\frac{1}{M_c}\sum_{i=1}^{M_c} p_{\text{nlos}}^i \log(\hat{p}_{\text{nlos}}^i) + (1 - p_{\text{nlos}}^i)\log(1 - \hat{p}_{\text{nlos}}^i), \tag{2}$$

where p_{nlos}^i denotes the true label corresponding to the i-th CIR sample in the data set and is one if the CIR corresponds to an NLoS link and zero otherwise. Furthermore, M_c represents the total number of CIRs in the training set. The formulation in (2) is also known in the literature as the binary cross-entropy loss function. The goal of training is then to adjust the weights of the neurons such that (2) is minimized. Finally, when the

trained DNN is employed in the context of joint sync&loc algorithm, the decision on the link condition is fed into the algorithm using the binary parameter ζ_i, which is set to one when $\hat{p}_{\text{nlos}} > 0.5$ and zero otherwise. Specifically, if ζ_i is zero, the communication link is considered NLoS and any measurement corresponding to it, i.e., time-stamp exchange and AoA, is dropped.

In the sequel, we present the principles of the AoA estimation algorithm, which draws on the CIRs employed for NLoS identification.

D. Angle of Arrival

The CIR fed into the DNN to identify the link condition can be treated as an input signal to the MUSIC algorithm to obtain the AoA. We present the principles of AoA estimation for UPAs based on [42]–[44]. The estimated AoA is given by

$$(\varphi_{ij}, \alpha_{ij}) = \arg\max_{\varphi,\alpha} \frac{1}{\mathbf{a}_n(\varphi,\alpha)^H \mathbf{N}\mathbf{N}^H \mathbf{a}_n(\varphi,\alpha)}, \tag{3}$$

where φ_{ij} and α_{ij} are the azimuth and elevation AoA of the signal received from the MU i at AP j, respectively. Parameter $\mathbf{a}_n(\varphi,\alpha)$ is the signal vector rotation on the n-th subcarrier and is given by (4), as shown at the bottom of the page, where N_{ant} denotes the number of AP antennas in one row (or column). Matrix $\mathbf{N}$ is constructed by $N_{\text{ant}}^2 - 1$ most right columns of the eigenvectors obtained when performing the eigen decomposition of the covariance matrix of the received signal. That is

$$\mathbf{R} = \mathbf{V}\mathbf{A}\mathbf{V}^H, \tag{5}$$

where matrices $\mathbf{A}$ and $\mathbf{V}$ contain the eigenvalues and eigenvectors, respectively. Furthermore,

$$\mathbf{R} = \frac{1}{N_s}\sum_{n=1}^{N_s} \mathbf{x}_n \mathbf{x}_n^H, \tag{6}$$

where the vector $\mathbf{x}_n$ is of dimension $N_{\text{ant}}^2 \times 1$ and represents the n-th element of FFT of the CIRs. The number of subcarriers, or, alternatively, the size of FFT is denoted by N_s. It is worth mentioning that, when constructing $\mathbf{N}$, the eigen decomposition in (5) is assumed to sort the eigenvalues in decreasing order. Lastly, each AP is assumed to have N_{ant}^2 CIRs at its disposal.

III. Clock Parameters and Position Estimation

In this section, we discuss an estimation method for the clock and position parameters. It relies primarily on the components analyzed in the previous section, i.e, time-stamp exchange, AoA estimation, and NLoS identification. In particular, given Section II-B2, and considering AP j as the master node, we can write

$$\frac{1}{\tilde{\gamma}_i}(c_i(t_2^k) - \tilde{\theta}_i) = c_j(t_1^k) + \frac{d_{ij}}{v_c} + T_{ij}^{k,0}, \tag{7}$$

$$\frac{1}{\tilde{\gamma}_i}(c_i(t_4^k) - \tilde{\theta}_i) = c_j(t_3^k) + \frac{d_{ij}}{v_c} + T_{ij}^{k,1}, \tag{8}$$

$$\frac{1}{\tilde{\gamma}_i}(c_i(t_5^k) - \tilde{\theta}_i) = c_j(t_6^k) - \frac{d_{ij}}{v_c} - R_{ij}^k, \tag{9}$$

where t_1^k/t_2^k, t_3^k/t_4^k, and t_5^k/t_6^k are the time points where MU i and AP j send/receive the sync messages, respectively. Parameter $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ denotes the Euclidean distance between nodes i and j. We note that, in Figure 3, instead of a global time reference $c(t) = t$, we take node j as the master node. It is straightforward to see that $\frac{1}{\tilde{\gamma}_i} = \frac{\gamma_j}{\gamma_i}$, $\tilde{\theta}_i = \theta_i - \tilde{\gamma}_i\theta_j$, $\tilde{d}_{ij} + \tilde{T}_{ij}^k = \gamma_j(d_{ij} + T_{ij}^k)$, and $\tilde{d}_{ij} - \tilde{R}_{ij}^k = \gamma_j(d_{ij} - R_{ij}^k)$. For the sake of simplicity, as done in [1], we assume $\tilde{d}_{ij} = d_{ij}$, $\tilde{R}_{ij}^k = R_{ij}^k$, and $\tilde{T}_{ij}^k = T_{ij}^k$. This is valid because $\gamma_j \approx 1$ and the values of $d_{ij} + T_{ij}^k$ and $d_{ij} - R_{ij}^k$ are small. In what follows, we first give the probabilistic representation of the problem. Subsequently, the principles of the estimation method are presented.

A. Probabilistic Formulation of the Problem

Let $\boldsymbol{\xi}_i^k$ be the state of the vector variable $\boldsymbol{\xi}_i \triangleq \begin{bmatrix}\tilde{\boldsymbol{\vartheta}}_i & \mathbf{p}_i\end{bmatrix}^T$ after the k-th round of time-stamp exchange, where $\tilde{\boldsymbol{\vartheta}}_i = \begin{bmatrix}\frac{1}{\tilde{\gamma}_i} & \frac{\tilde{\theta}_i}{\tilde{\gamma}_i}\end{bmatrix}$ and $\mathbf{p}_i = \begin{bmatrix}x_i & y_i\end{bmatrix}$. Parameters x_i and y_i denote the position of node i on the x and y axes, respectively. The aim is then to infer the pdf corresponding to the k-th state, which can be written as

$$\begin{aligned} &p(\boldsymbol{\xi}_i^k | \{\mathbf{c}_{ij}^{1:k}, \varphi_{ij}^{1:k}, \zeta_{ij}^{1:k}\}_{\forall j \in \mathcal{I}_i}) \\ &= \int p(\boldsymbol{\xi}_i^0, \cdots, \boldsymbol{\xi}_i^k | \{\mathbf{c}_{ij}^{1:k}, \varphi_{ij}^{1:k}, \zeta_{ij}^{1:k}\}_{\forall j \in \mathcal{I}_i}) d\boldsymbol{\xi}_i^0 \cdots d\boldsymbol{\xi}_i^{k-1}, \end{aligned} \tag{10}$$

where the superscript $1{:}k$ indicates the collection of measurements from the first round until the k-th. Applying Bayes rule, we can rewrite (10) as

$$\begin{aligned} &p(\boldsymbol{\xi}_i^k | \{\mathbf{c}_{ij}^{1:k}, \varphi_{ij}^{1:k}, \zeta_{ij}^{1:k}\}_{\forall j \in \mathcal{I}_i}) \\ &\propto \int p(\{\mathbf{c}_{ij}^{1:k}, \varphi_{ij}^{1:k}, \zeta_{ij}^{1:k}\}_{\forall j \in \mathcal{I}_i} | \boldsymbol{\xi}_i^0, \cdots, \boldsymbol{\xi}_i^k) \\ &\quad \times p(\boldsymbol{\xi}_i^0, \cdots, \boldsymbol{\xi}_i^k) d\boldsymbol{\xi}_i^0 \cdots d\boldsymbol{\xi}_i^{k-1}. \end{aligned} \tag{11}$$

Figure 5 depicts the temporal evolution of $\boldsymbol{\xi}_i^k$ as well as its relation with the measurements at each time step. Such a structure is referred to as dynamic Bayesian Network (BN), in which a basic BN repeats itself in each time step [45]. The states of a dynamic BN, i.e., all the variables with the same

$$\begin{aligned} \mathbf{a}_n(\varphi,\alpha) = \Big[1,\ & e^{i\frac{2\pi d}{\lambda}\sin(\alpha)(\sin(\varphi)+\cos(\varphi))},\ e^{i\frac{2\pi d}{\lambda}\sin(\alpha)(\sin(\varphi)+2\cos(\varphi))},\ \cdots, \\ & e^{i\frac{2\pi d}{\lambda}\sin(\alpha)((N_{\text{ant}}-1)\sin(\varphi)+(N_{\text{ant}}-2)\cos(\varphi))},\ e^{i\frac{2\pi d}{\lambda}(N_{\text{ant}}-1)\sin(\alpha)(\sin(\varphi)+\cos(\varphi))}\Big]^T_{1\times N_{\text{ant}}^2} \end{aligned} \tag{4}$$

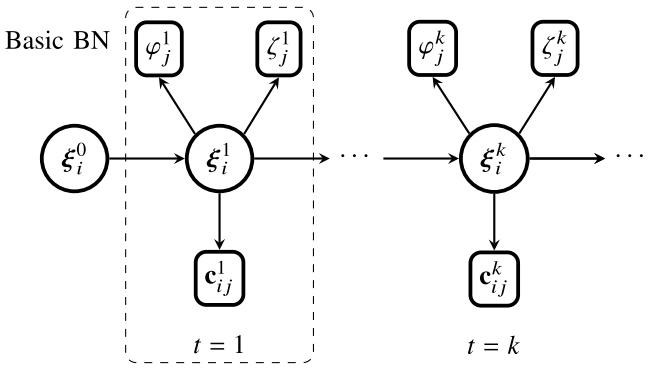


Fig. 5. Dynamic Bayesian network representing the temporal evolution of the vector variable $\boldsymbol{\xi}_i$ and its relation to the measurements. $\boldsymbol{\xi}_i^k$ is obtained through time-stamp exchange while φ_j^k and ζ_j^k are estimated with the aid of CIR.

time index, satisfy the Markov property[1] [46], enabling us to carry out the following mathematical simplifications. In particular, knowing that the measurements are independent and assuming the Markov property, we reformulate the integrands in (11) as

$$\begin{aligned} &p(\{\mathbf{c}_{ij}^{1:k},\varphi_{ij}^{1:k},\zeta_{ij}^{1:k}\}_{\forall j\in\mathcal{I}_i}|\boldsymbol{\xi}_i^0,\cdots,\boldsymbol{\xi}_i^k)\\ &= p(\{\mathbf{c}_{ij}^{k},\varphi_{ij}^{k},\zeta_{ij}^{k}\}_{\forall j\in\mathcal{I}_i}|\boldsymbol{\xi}_i^k)\cdots p(\{\mathbf{c}_{ij}^{1},\varphi_{ij}^{1},\zeta_{ij}^{1}\}_{\forall j\in\mathcal{I}_i}|\boldsymbol{\xi}_i^1),\\ &\quad p(\boldsymbol{\xi}_i^0,\cdots,\boldsymbol{\xi}_i^k) = p(\boldsymbol{\xi}_i^k|\boldsymbol{\xi}_i^{k-1})\cdots p(\boldsymbol{\xi}_i^1|\boldsymbol{\xi}_i^0)p(\boldsymbol{\xi}_i^0), \end{aligned} \tag{12}$$

where $p(\boldsymbol{\xi}_i^0)$ denotes the prior knowledge on $\boldsymbol{\xi}_i$. Plugging (12) into (11) leads to

$$\begin{aligned} &p(\boldsymbol{\xi}_i^k|\mathbf{c}_{ij}^{1:k},\varphi_{ij}^{1:k},\zeta_{ij}^{1:k})\\ &\propto \underbrace{\int p(\boldsymbol{\xi}_i^0)\left[\prod_{r=1}^{k-1}p(\boldsymbol{\xi}_i^r|\boldsymbol{\xi}_i^{r-1})p(\mathbf{c}_{ij}^r,\varphi_{ij}^r,\zeta_{ij}^r|\boldsymbol{\xi}_i^r)\right]p(\boldsymbol{\xi}_i^k|\boldsymbol{\xi}_i^{k-1})}_{=p(\boldsymbol{\xi}_i^k|\mathbf{c}_{ij}^{1:k-1},\varphi_{ij}^{1:k-1},\zeta_{ij}^{1:k-1})}\\ &\quad\times p(\mathbf{c}_{ij}^k,\varphi_{ij}^k,\zeta_{ij}^k|\boldsymbol{\xi}_i^k)d\boldsymbol{\xi}_i^0\cdots d\boldsymbol{\xi}_i^{k-1}. \end{aligned} \tag{13}$$

Finally, we can write

$$\begin{aligned} &p(\boldsymbol{\xi}_i^k|\{\mathbf{c}_{ij}^{1:k},\varphi_{ij}^{1:k},\zeta_{ij}^{1:k}\}_{\forall j\in\mathcal{I}_i})\\ &\propto p(\boldsymbol{\xi}_i^k|\{\mathbf{c}_{ij}^{1:k-1},\varphi_{ij}^{1:k-1},\zeta_{ij}^{1:k-1}\}_{\forall j\in\mathcal{I}_i})p(\mathbf{c}_{ij}^k,\varphi_{ij}^k,\zeta_{ij}^k|\boldsymbol{\xi}_i^k). \end{aligned} \tag{14}$$

The term $p(\boldsymbol{\xi}_i^k|\{\mathbf{c}_{ij}^{1:k-1},\varphi_{ij}^{1:k-1},\zeta_{ij}^{1:k-1}\}_{\forall j\in\mathcal{I}_i})$ is referred to as *prediction* step while the term $p(\mathbf{c}_{ij}^k,\varphi_{ij}^k,\zeta_{ij}^k|\boldsymbol{\xi}_i^k)$ is considered as *correction* step [46]. If the Gaussian assumption about $\boldsymbol{\xi}_i^0$ held and the relation between all the states in Figure 5 were linear, we could conclude that the marginal in (14) would also be Gaussian distributed. Unfortunately, that is not the case in the joint sync&loc problem as the measurement equations (and consequently the correction steps) are partially non-linear. In concrete terms, the aforementioned problem stems from the nonlinear relation between the location parameters (x_i, y_i) and the time-stamps in (7), (8), (9) on one hand, and the measured AoA in (3) on the other hand.

[1]It postulates that the state of the system at time t depends only on its immediate past, i.e. its state at time $t-1$.

There are several approaches to tackle the nonlinearity problem and, consequently, to estimate the non-Gaussian posterior distribution. In [14], it is proposed to undertake the Taylor expansion of the nonlinear terms around the prediction point, while [8], [9], [47] have employed EKF to address the non-linearity. In addition to being prone to divergence, which is hard to mitigate analytically, all of these methods require initialization and even then are only able to deliver medium accuracy. In what follows, we discuss the details of a novel joint sync&loc approach based on PGM filters.

B. Particle Gaussian Mixure Filter

The idea underpinning PGM filters is to approximate a pdf by the sum of weighted *Gaussian density functions* (gdfs) [17]. Leveraging this idea, we can write the posterior in (14) as

$$p(\boldsymbol{\xi}_i^k|\{\mathbf{c}_{ij}^{1:k},\varphi_{ij}^{1:k},\zeta_{ij}^{1:k}\}_{\forall j\in\mathcal{I}_i}) = \sum_{f=1}^{F} w_f^k\mathcal{N}(\boldsymbol{\xi}_i^k|\boldsymbol{\mu}_f^k,\boldsymbol{\Sigma}_f^k), \tag{15}$$

with $\sum_{f=1}^{F} w_f^k = 1, w_f^k \geq 0 \forall f$, where $\boldsymbol{\mu}_f^k = \begin{bmatrix}\boldsymbol{\mu}(\tilde{\boldsymbol{\vartheta}}_i)_f^k & \boldsymbol{\mu}(\mathbf{p}_i)_f^k\end{bmatrix}$ and $\boldsymbol{\Sigma}_f^k = \begin{bmatrix}\boldsymbol{\Sigma}(\tilde{\boldsymbol{\vartheta}}_i)_f^k & \mathbf{0}_2\\ \mathbf{0}_2 & \boldsymbol{\Sigma}(\mathbf{p}_i)_f^k\end{bmatrix}$ denote the mean vector and covariance matrix of the f-th gdf in the k-th round of estimation, respectively. Parameter F represents the total number of gdfs. Furthermore, $\boldsymbol{\mu}(\tilde{\boldsymbol{\vartheta}}_i)_f^k/\boldsymbol{\mu}(\mathbf{p}_i)_f^k$ and $\boldsymbol{\Sigma}(\tilde{\boldsymbol{\vartheta}}_i)_f^k/\boldsymbol{\Sigma}(\mathbf{p}_i)_f^k$ represent the mean vector and covariance matrix corresponding to the vector variable $\tilde{\boldsymbol{\vartheta}}_i/\mathbf{p}_i$, respectively.

Seeking to further simplify (15), we reformulate (7), (8), and (9) as follows. Subtracting (7) from (8) leads to

$$\frac{1}{\tilde{\gamma}_i}(c_i(t_4^k)-c_i(t_2^k)) = c_j(t_3^k)-c_j(t_1^k)+T_{ij}^{k,1}-T_{ij}^{k,0}, \tag{16}$$

while summing up (8) and (9) gives

$$\frac{1}{\tilde{\gamma}_i}(c_i(t_4^k)+c_i(t_5^k)-2\tilde{\theta}_i) = c_j(t_3^k)+c_j(t_6^k)+T_{ij}^{k,1}-R_{ij}^k. \tag{17}$$

It is straightforward to observe that $\tilde{\boldsymbol{\vartheta}}_i^k$, on one hand, is linearly dependent on the time-stamps, and, on the other hand, does not depend on $\mathbf{p}_i$. This suggests that, although the $p(\mathbf{c}_{ij}^k,\varphi_{ij}^k,\zeta_{ij}^k|\boldsymbol{\xi}_i^k)$ is not Gaussian distributed in general, it is indeed Gaussian across the $\tilde{\boldsymbol{\vartheta}}_i$ axis as both T_{ij} and R_{ij} are Gaussian distributed. We capitalize on the linear Gaussian substructures in the model to keep the state dimensions low. Consequently, the gdfs can be employed only across the $\mathbf{p}_i$ axis transforming the structure of (15) into the multiplication of a single gdf across $\tilde{\boldsymbol{\vartheta}}_i$ and sum weighted of multiple gdfs across $\mathbf{p}_i$ (visualized in Figure 6). Such a structure not only lays the ground for the hybrid parametric and particle-based implementation of BRF-based joint sync&loc estimation but also dramatically reduces the computational burden. Given above, (15) can be simplified as

$$\begin{aligned} &p(\{\mathbf{c}_{ij}^{1:k},\varphi_{ij}^{1:k},\zeta_{ij}^{1:k}\}_{\forall j\in\mathcal{I}_i}|\boldsymbol{\xi}_i^k)\\ &= \mathcal{N}(\tilde{\boldsymbol{\vartheta}}_i^k|\boldsymbol{\mu}(\tilde{\boldsymbol{\vartheta}}_i)^k,\boldsymbol{\Sigma}(\tilde{\boldsymbol{\vartheta}}_i)^k)\times\sum_{f=1}^{F} w_f^k\mathcal{N}(\mathbf{p}_i^k|\boldsymbol{\mu}(\mathbf{p}_i)_f^k,\boldsymbol{\Sigma}(\mathbf{p}_i)_f^k). \end{aligned} \tag{18}$$

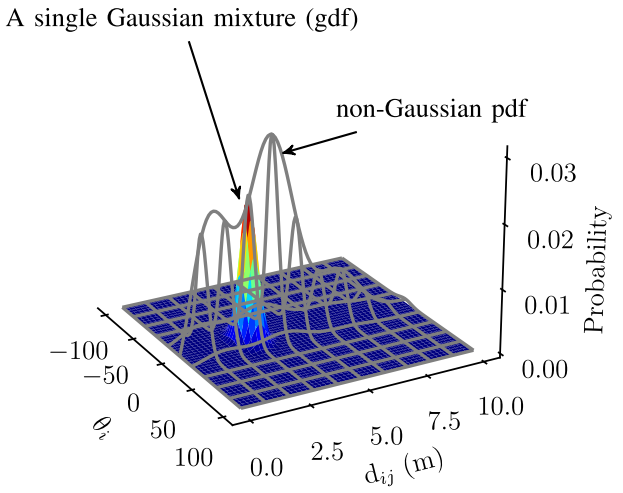


Fig. 6. An example distribution of the $\boldsymbol{\xi}_i$ for a given time-stamp measurement.

We note that when $\boldsymbol{\Sigma}(\mathbf{p}_i)_f^k$ approaches 0, the term $\mathcal{N}(\mathbf{p}_i^k|\boldsymbol{\mu}(\mathbf{p}_i)_f^k, \boldsymbol{\Sigma}(\mathbf{p}_i)_f^k)$ tends towards $\delta(\mathbf{p}_i^k - \boldsymbol{\mu}(\mathbf{p}_i)_f^k)$, where $\delta(\cdot)$ denote the Dirac impulse function. Such a function forms the basis of the classical particle filter. In what follows, we further delve into the steps of parameter estimation of the above-mentioned distribution. Firstly, the details of *prediction* step are described, where all the parameters are denoted by $(\cdot)_-$. Next, we obtain the likelihood of the measurements whose parameters are represented by $(\cdot)_+$. Lastly, we compute the parameters of the posterior distribution in (18) and perform the resampling.

1) Prediction: Given the linear dynamics of MUs' clocks and movements, a reasonable prediction for $\boldsymbol{\xi}_i^k$ is given by

$$\begin{aligned} &p(\boldsymbol{\xi}_i^k|\{\mathbf{c}_{ij}^{1:k-1}, \varphi_{ij}^{1:k-1}, \zeta_{ij}^{1:k-1}\}_{\forall j\in\mathcal{I}_i}) \\ &= \mathcal{N}(\tilde{\boldsymbol{\vartheta}}_i^k|\boldsymbol{\mu}(\tilde{\boldsymbol{\vartheta}}_i)_-^k, \boldsymbol{\Sigma}(\tilde{\boldsymbol{\vartheta}}_i)_-^k) \\ &\quad\times \sum_{f=1}^{F} w_{f-}^k \mathcal{N}\left(\mathbf{p}_i^k|\boldsymbol{\mu}(\mathbf{p}_i)_{f-}^k, \boldsymbol{\Sigma}(\mathbf{p}_i)_{f-}^k\right), \end{aligned} \tag{19}$$

where

$$w_{f-}^k = \frac{1}{F}\mathbf{1}_F, \quad \boldsymbol{\mu}(\mathbf{p}_i)_{f-}^k = \boldsymbol{\mu}(\mathbf{p}_i)_f^{k-1} + \mathbf{n}_f,$$

with $\mathbf{n}_f$ being the noise vector derived from the distribution $\mathcal{N}(\mathbf{n}|\mathbf{0}, \mathbf{Q}_n(\mathbf{p}_i))$, for $\mathbf{Q}_n(\mathbf{p}_i) = \text{diag}(\sigma_x^2, \sigma_y^2)$. In practice, we initialize $\boldsymbol{\Sigma}(\mathbf{p}_i)_{f-}^k \propto \text{diag}(F^{-0.4}, F^{-0.4})$, which is proved in [16] to be the optimal choice. Furthermore, according to [12],

$$\boldsymbol{\mu}(\tilde{\boldsymbol{\vartheta}}_i)_-^k = \mathbf{F}\boldsymbol{\mu}(\tilde{\boldsymbol{\vartheta}}_i)^{k-1} + \mathbf{u}, \tag{20}$$

$$\boldsymbol{\Sigma}(\tilde{\boldsymbol{\vartheta}}_i)_-^k = \mathbf{F}\boldsymbol{\Sigma}(\tilde{\boldsymbol{\vartheta}}_i)^{k-1}\mathbf{F}^T + \mathbf{Q}_n(\tilde{\boldsymbol{\vartheta}}_i), \tag{21}$$

with

$$\mathbf{F} = \begin{bmatrix} 1 & 0 \\ T & 1 \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} 0 \\ T \end{bmatrix}, \quad \mathbf{Q}_n(\tilde{\boldsymbol{\vartheta}}_i) = \text{diag}(\sigma_\gamma^2, \sigma_\theta^2).$$

The matrices $\mathbf{Q}_n(\tilde{\boldsymbol{\vartheta}}_i)$ and $\mathbf{Q}_n(\mathbf{p}_i)$ denote the covariance of the zero-mean Gaussian noises on each gdf across the $\tilde{\boldsymbol{\vartheta}}_i$ and $\mathbf{p}_i$ axes, respectively. In general, the design of $\mathbf{Q}_n(\cdot)$ is a difficult task. In particular, if it is too small, the filter will be overconfident in its prediction model and will diverge from the actual solution. In contrast, if it is too large, it will be unduly dominated by the noise in the measurements and perform sub-optimally [48]. Similar to [1], [37], [39], we set σ_γ^2 and σ_θ^2, such that the external noises as well as the residues from the previous iteration are accounted for. Furthermore, to determine the value of σ_x^2 and σ_y^2, the design model discussed in [47], [48] is followed. That is, opting for a noise variance that is large enough to allow the gdfs to assign a reasonable probability to the locations where the MU might be. In the urban scenario, for example, the maximum permitted speed is 50 km/h (≈14 m/s), resulting in $\sigma_x = \sigma_y = 14 \times T$.

2) Measurement Likelihood and Weight Update: The same structure as (18) is imposed on the likelihood of the measurements. That is,

$$\begin{aligned} &p(\mathbf{c}_{ij}^k, \varphi_{ij}^k, \zeta_{ij}^k|\boldsymbol{\xi}_i^k) \\ &= \mathcal{N}(\tilde{\boldsymbol{\vartheta}}_i^k|\boldsymbol{\mu}(\tilde{\boldsymbol{\vartheta}}_i)_+^k, \boldsymbol{\Sigma}(\tilde{\boldsymbol{\vartheta}}_i)_+^k) \\ &\quad\times \sum_{f=1}^{F} w_{f+}^k \mathcal{N}\left(\mathbf{p}_i^k|\boldsymbol{\mu}(\mathbf{p}_i)_{f+}^k, \boldsymbol{\Sigma}(\mathbf{p}_i)_{f+}^k\right). \end{aligned} \tag{22}$$

To obtain the parameters of the above likelihood, we firstly transform (16) and (17) into the matrix form. That is,

$$\mathbf{B}_{ij}^k \tilde{\boldsymbol{\vartheta}}_i^k = \mathbf{r}_{ij}^k + \mathbf{z}_{ij}, \tag{23}$$

where $\mathbf{z}_{ij} \sim \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{R}_{ij}^k)$ with $\mathbf{R}_{ij}^k = \text{diag}(2\sigma_{T_{ij}}^2, \sigma_{T_{ij}}^2 + \sigma_{R_{ij}}^2)$, and

$$\mathbf{B}_{ij}^k = \begin{bmatrix} c_i(t_4^k) - c_i(t_2^k) & 0 \\ c_i(t_4^k) + c_i(t_5^k) & -2 \end{bmatrix}, \quad \mathbf{r}_{ij}^k = \begin{bmatrix} c_j(t_3^k) - c_j(t_1^k) \\ c_j(t_3^k) + c_j(t_6^k) \end{bmatrix}.$$

The mean and covariance matrix of the gdfs across the $\tilde{\boldsymbol{\vartheta}}_i$ axis can be written as

$$\boldsymbol{\mu}(\tilde{\boldsymbol{\vartheta}}_i)_+^k = \mathbf{A}_{ij}^k \mathbf{r}_{ij}^k, \quad \boldsymbol{\Sigma}(\tilde{\boldsymbol{\vartheta}}_i)_+^k = \mathbf{A}_{ij}^k \mathbf{R}_{ij}^k (\mathbf{A}_{ij}^k)^T, \tag{24}$$

where $\mathbf{A}_{ij}^k = ((\mathbf{B}_{ij}^k)^T \mathbf{B}_{ij}^k)^{-1} (\mathbf{B}_{ij}^k)^T$.

To obtain the location parameters corresponding to each gdf, we can assume that the measurement equations are linear in the vicinity of the point predicted by the prediction step. That is, to approximate them with their first-order Taylor expansions, the details of which are thoroughly explained in [14], [15].[2] The measurement equations we rely on to estimate the parameters of the likelihoods are (9) and

$$\arctan\left(\frac{y_i - y_j}{x_i - x_j}\right) = \varphi_{ij}^k, \tag{25}$$

where φ_{ij}^k is calculated as explained in Section II-D. Carrying out the necessary mathematical manipulation, we can write the same relation as (23) for each gdf. That is,

$$\mathbf{B}_{ij,f}^k \mathbf{p}_i^k = \mathbf{r}_{ij,f}^k + \mathbf{z}_{ij,f}, \tag{26}$$

where $\mathbf{z}_{ij,f} \sim \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{R}_{ij,f})$ with $\mathbf{R}_{ij,f} = \text{diag}(\sigma_{R_{ij}}^2, \sigma_\varphi^2)$. Furthermore, $\mathbf{B}_{ij,f} = \begin{bmatrix} \mathbf{a}_j^k & \mathbf{b}_j^k \end{bmatrix}^T$ with the vectors $\mathbf{a}_j^k$ and $\mathbf{b}_j^k$

[2]Note that this is equivalent to EKF, e.g. that of [47]. Nevertheless, to keep consistency with the approach taken in this work, i.e., Bayesian representation of the filtering process, we avoid the EKF representation.

calculated by means of (27) and (28), as shown at the bottom of the page, respectively. Finally, $\mathbf{r}_{ij,f}$ is constructed as in (29), as shown at the bottom of the page. We note that (27) and (28) are computed by means of the Taylor expansion of (9) and (25) around the predicted point $\boldsymbol{\mu}(\mathbf{p}_i)^k_{f-}$ with the known $\boldsymbol{\mu}(\tilde{\boldsymbol{\vartheta}}_i)^k_+$ obtained by (24). Given (26), and similar to (24), we can write

$$\begin{aligned}\boldsymbol{\mu}(\mathbf{p}_i)^k_{f+} &= \mathbf{A}^k_{ij,f}\mathbf{r}^k_{ij,f},\\ \boldsymbol{\Sigma}(\mathbf{p}_i)^k_{f+} &= \mathbf{A}^k_{ij,f}\mathbf{R}^k_{ij,f}(\mathbf{A}^k_{ij,f})^T,\end{aligned} \tag{30}$$

where $\mathbf{A}^k_{ij,f} = ((\mathbf{B}^k_{ij,f})^T\mathbf{B}^k_{ij,f})^{-1}(\mathbf{B}^k_{ij,f})^T$. Furthermore, it is straightforward to see that

$$w^k_{f+} = \mathcal{N}(\mathbf{p}^k_i = \boldsymbol{\mu}(\mathbf{p}_i)^k_{f+}|\boldsymbol{\mu}(\mathbf{p}_i)^k_{f+}, \boldsymbol{\Sigma}(\mathbf{p}_i)^k_{f+}). \tag{31}$$

In other words, the weights are equal to the likelihood of the mean of each gdf.

3) Posterior Estimation: Having taken the necessary steps, we can now compute (18) as an approximation for the posterior distribution in (14). Multiplying (19) and (22), the parameters of (18) can be given by

$$\begin{aligned}\boldsymbol{\mu}(\tilde{\boldsymbol{\vartheta}}_i)^k &= \left[\boldsymbol{\Sigma}(\tilde{\boldsymbol{\vartheta}}_i)^k_- + \boldsymbol{\Sigma}(\tilde{\boldsymbol{\vartheta}}_i)^k_+\right]^{-1}\\ &\quad\times\left(\boldsymbol{\Sigma}(\tilde{\boldsymbol{\vartheta}}_i)^k_+\boldsymbol{\mu}(\tilde{\boldsymbol{\vartheta}}_i)^k_- + \boldsymbol{\Sigma}(\tilde{\boldsymbol{\vartheta}}_i)^k_-\boldsymbol{\mu}(\tilde{\boldsymbol{\vartheta}}_i)^k_+\right),\end{aligned} \tag{32}$$

$$\boldsymbol{\Sigma}(\tilde{\boldsymbol{\vartheta}}_i)^k = \left[\left(\boldsymbol{\Sigma}(\tilde{\boldsymbol{\vartheta}}_i)^k_-\right)^{-1} + \left(\boldsymbol{\Sigma}(\tilde{\boldsymbol{\vartheta}}_i)^k_+\right)^{-1}\right]^{-1}. \tag{33}$$

The final estimation of the clock skew and offset can then be given by

$$\tilde{\gamma}^k_i = \frac{1}{\boldsymbol{\mu}(\tilde{\boldsymbol{\vartheta}}_i)^k[1]}, \quad \tilde{\theta}^k_i = \frac{\boldsymbol{\mu}(\tilde{\boldsymbol{\vartheta}}_i)^k[2]}{\boldsymbol{\mu}(\tilde{\boldsymbol{\vartheta}}_i)^k[1]}. \tag{34}$$

Furthermore, each gdf can be updated across $\mathbf{p}_i$ axis by

$$\begin{aligned}\boldsymbol{\mu}(\mathbf{p}_i)^k_f &= \left[\boldsymbol{\Sigma}(\mathbf{p}_i)^k_{f-} + \boldsymbol{\Sigma}(\mathbf{p}_i)^k_{f+}\right]^{-1}\\ &\quad\times\left(\boldsymbol{\Sigma}(\mathbf{p}_i)^k_{f+}\boldsymbol{\mu}(\mathbf{p}_i)^k_{f-} + \boldsymbol{\Sigma}(\mathbf{p}_i)^k_{f-}\boldsymbol{\mu}(\mathbf{p}_i)^k_{f+}\right),\end{aligned} \tag{35}$$

$$\boldsymbol{\Sigma}(\mathbf{p}_i)^k_f = \left[\left(\boldsymbol{\Sigma}(\mathbf{p}_i)^k_{f-}\right)^{-1} + \left(\boldsymbol{\Sigma}(\mathbf{p}_i)^k_{f+}\right)^{-1}\right]^{-1}. \tag{36}$$

Next, the weights can be updated as

$$w^k_f = \frac{w^k_{f-}w^k_{f+}}{\sum_{f=1}^F w^k_{f-}w^k_{f+}}. \tag{37}$$

Given (35), (36), (37), the final position estimation can be given by

$$\hat{\mathbf{p}_i}^k = \sum_{f=1}^F w^k_f\boldsymbol{\mu}(\mathbf{p}_i)^k_f. \tag{38}$$

4) Resampling and Tuning: Resampling is one of the most crucial steps when performing PGM filtering. Without the resampling step, the filter would suffer from sample depletion. That is, after a while all the gdfs but a few will have negligible weight. Consequently, the posterior will be approximated with only a few gdfs, leading to its underestimation. To overcome this shortcoming, in each iteration we replace the minor-weight gdfs with new ones whose means are sampled from the approximated posterior. The sample depletion can be monitored throughout the filtering process by calculating the number of effective gdfs as

$$N_{\text{eff}} = \frac{1}{\sum_{f=1}^F (w^k_f)^2}. \tag{39}$$

As can be seen, N_{eff} attains its maximum when all the weights are equal to $\frac{1}{F}$ and falls to its minimum when all but a single weight is equal to zero. In this work, the resampling is carried out when the $N_{\text{eff}} < \frac{2}{3}F$.

All above-mentioned steps are summarized in algorithm 1.

Algorithm 1 DePF Joint sync&loc

1: Initialize p($\boldsymbol{\xi}^0_i$) as in (18).
2: **for all** the APs in $\mathcal{I}_i$ **do**
3: Perform the time-stamp exchange mechanism described in Section II-B2 and Figure 3.
4: Estimate the CIR using QuaDRiGa channel model.
5: Estimate the AoA and the link condition ζ^k_{ij} using the CIR and (25)
6: **for all** LoS links ($\zeta^k_{ij} = 0$) **do**
7: Construct $\mathbf{B}^k_{ij}$, $\mathbf{B}^k_{ij,f}$, $\mathbf{R}^k_{ij}$, $\mathbf{R}^k_{ij,f}$, $\mathbf{r}^k_{ij}$ and $\mathbf{r}^k_{ij,f}$ by means of the time-stamps and the AoA.
8: Update the parameters of the posterior distribution using (32), (33), (35), and (36).
9: **end for**
10: **end for**
11: Estimate the clock and position parameters using (34) and (38).
12: **if** $N_{\text{eff}} < \frac{2}{3}F$ **then**
13: Perform resampling.
14: **end if**
15: Go to step 2.

C. Complexity of the Algorithm

The computational complexity of different types of BRF and PF filters including L-BRF and PGM has been extensively

$$a^k_{j,f} = \frac{1}{v_c}\left|\boldsymbol{\mu}(\mathbf{p}_i)^k_{f-} - \mathbf{p}_j\right|, \quad \mathbf{a}^k_{j,f} = \frac{1}{v_c^2 a^k_{j,f}}\left(\boldsymbol{\mu}(\mathbf{p}_i)^k_{f-} - \mathbf{p}_j\right), \tag{27}$$

$$b^k_{j,f} = \arctan(\frac{\mathbf{a}^k_{j,f}[2]}{\mathbf{a}^k_{j,f}[1]}), \quad \mathbf{b}^k_{j,f} = \frac{1}{a^k_{j,f}}\left[-\mathbf{a}^k_{j,f}[2],\, \mathbf{a}^k_{j,f}[1]\right]. \tag{28}$$

$$\mathbf{r}_{ij,f} = \left[c_j(t^k_6) - a^k_{j,f} + \boldsymbol{\mu}(\mathbf{p}_i)^k_{f-}\cdot\mathbf{a}^k_{j,f} - \begin{bmatrix}c_i(t^k_5) & -1\end{bmatrix}\cdot\boldsymbol{\mu}(\tilde{\boldsymbol{\vartheta}}_i)^k_+, \varphi^k_{ij} - b^k_{j,f} + \boldsymbol{\mu}(\mathbf{p}_i)^k_{f-}\cdot\mathbf{b}^k_{j,f}\right]^T \tag{29}$$

TABLE I
COMPLEXITY COMPARISON OF L-BRF AND PGM FILTER

	L-BRF	PGM
Prediction	$2l^3 + l^2$	$2l^3 + l^2$
Likelihood/Correction	$2l^3 + l^2$	$2Fn^3 + Fn^2 + 2l^3 + l^2$
Estimation	$7l^2$	$7Fn^2 + nF + 7l^2$
Total	$O(l^3)$	$O(Fn^3 + l^3)$

discussed in [18], [49]. Parameters l, n, and F denote the number of linear state variables, nonlinear state variables, and gdfs (or mixtures), respectively. For the sake of simplicity, we only consider the number of multiplications to evaluate the complexity. Table I shows the complexity for each step of L-BRF and PGM. For the prediction step, it can be seen from (19) and (21) that two squared matrix multiplications and a matrix-vector multiplication are needed. We note that the computation cost of generating random variables is $O(1)$. The same holds for the likelihood computation given in (24) and (30). In the PGM, however, the L-BRF is repeated F times for each gdf across the nonlinear state variables. For the estimation step, the L-BRF needs 4 matrix inversions and 3 matrix-vector multiplications. The same number of multiplications is necessary for each gdf of the PGM. This is in addition to the multiplications between the weights and the particles essential to obtain the final estimation. Finally, we need to perform a cumulative sum to perform resampling, whose complexity is considered to be $O(F)$. It is apparent that PGM adds an overhead, however, it turns out that, according to [18], [49], PGM is more efficient, especially when the uncertainty of the measurements increases.

IV. SIMULATION RESULTS AND DISCUSSION

In this section, we evaluate the performance of the techniques employed in this work. In particular, we first evaluate the performance of a DNN-based NLoS identifier. Next, we present the result of AoA estimation. Finally, the performance of the joint sync&loc algorithm developed in this work is thoroughly analyzed.

A. DNN-Based NLoS Identification

To perform NLoS identification, the DNN in Figure 4 needs to be trained first. The training data is obtained using the QuaDRiGa channel model, the details of which are given in [50]. Specifically, the MU's movement profile can be implemented under the Urban Micro (UMi) cell scenario (denoted by "3GPP_38.901_UMi" in the QuaDRiga documentation), which corresponds to the densely populated urban areas. We collect 5000 CIR realizations for each scenario, i.e., LoS and NLoS, 80% of which is used for the training purpose while the remaining 20% is treated as the test set. To prepare the CIRs to be fed into the DNN, we first input them into a 64-point FFT to obtain the Channel Frequency Responses (CFRs). Subsequently, we take the magnitude of the CFRs and normalize each to its maximum component so that all magnitudes are between 0 and 1. Such normalization is proved to result in faster learning and convergence [51].

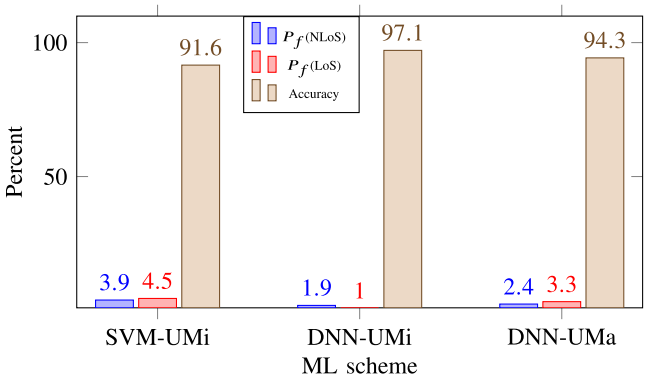


Fig. 7. Comparison of two ML schemes when performing NLoS-identification. P_f(LoS)/P_f(NLoS) denotes the probability that the true condition of the links detected as LoS/NLoS is NLoS/LoS.

The normalized magnitudes of the CFRs are then fed into a DNN with 2 hidden layers, each comprising 50 neurons with a rectified linear unit activation function. The loss function in (2) is then optimized using Adam optimizer to obtain the weights of each neuron. Furthermore, the probability that a CFR corresponds to a LoS and NLoS link condition is indicated by the DNN's two output neurons with a softmax activation function.

Figure 7 depicts the accuracy of the NLoS-identifier based on SVM, a classical ML algorithm, and DNN, the method proposed in this work. As can be seen, the DNN-based method delivers higher accuracy, outperforming the classical method. Specifically, DNNs are more powerful when it comes to estimating the classifier function, and, therefore, they turn in superior performance. The performance remains high even if we employ the DNN in an environment other than that of the training data, i.e., Urban Macro (UMa) cells instead of UMi. If the environment is too dissimilar, the performance will drastically deteriorate. In our simulations, we observed a poor accuracy of 61% for the rural-urban scenario, which is highly different from the UMi or UMa.

As mentioned before, the extremely high accuracy provided by the DNN is crucial as determining the link condition is among the most important decisions to be taken. In particular, false detection of NLoS links as LoS, P_f(LoS), not only can result in a poor estimation of the MU position and clock parameters, but also may lead to divergence of the filter. This occurs since the AoA estimation as well as the time-based distance measurement (which in the case of this work is carried out through time-stamp exchange) are highly inaccurate for NLoS links.

B. AoA Estimation

To evaluate the performance of the MUSIC algorithm, we arrange a specific simulation setup (shown in Figure 8) where an MU moves with the velocity of 2 m/s along the x axis from the point $[x = 0, y = 0, z = 1.5]$ until $[x = 70, y = 0, z = 1.5]$. An AP with a $N_{\text{ant}} \times N_{\text{ant}}$ UPA and tilted 20° is located at $[x = 35, y = -5, z = 10]$, equally distant from the two edges of the trajectory. Figure 8 depicts such a setup where the MU's trajectory and AP's coverage area

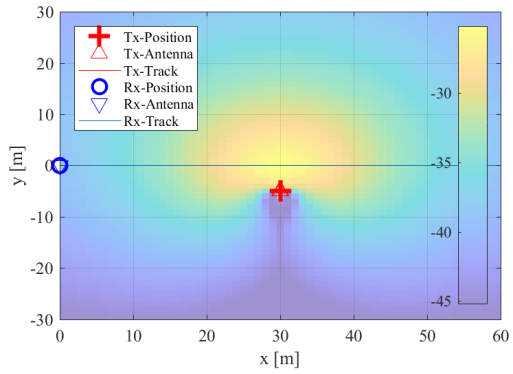


Fig. 8. Simulation setup for calculating the AoA.

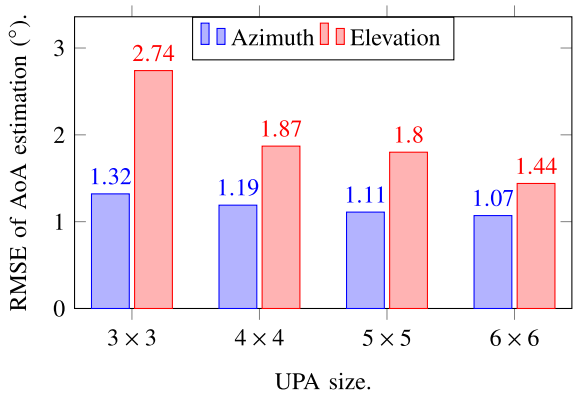


Fig. 9. AoA estimation accuracy.

(for 23 dBm power allocated to each antenna element) are observable. Furthermore, the elements are assumed to be patch antennas with 90° and 180° beam opening in the elevation and azimuth plane, respectively. Such a setup covers all possible angles that an MU might have with respect to an AP, i.e., from 6° to 171°. Furthermore, it represents the basic movement of the MUs in an urban scenario, e.g., the movement profile of the users shown in Figure 1 can be seen as the combination of that depicted in Figure 8. Lastly, at each time step, the AoA is estimated using the MUSIC algorithm fed with the corresponding CIR generated by QuaDRiGa. The algorithm estimates the azimuth and elevation AoA using the binary exhaustive search up to the 0.5 degree accuracy level, where the number of search bins are 40 and 20, respectively.

Figure 9 depicts the Root Mean Square Error (RMSE) of the AoA estimation for several UPA sizes. As can be observed, the RMSE of azimuth AoA estimation remains under 1.5° for almost all the investigated UPAs, which paves the way for precise localization of the MUs. Nevertheless, in our simulations, we observed that for smaller UPAs the RMSE increases drastically due to the large errors at the edges of the trajectory. Although such cases rarely occur, they can potentially lead to filter divergence. Moreover, the same behavior is observed for the elevation AoA estimation. Generally, as can be seen in the figure, the RMSE is slightly higher for the elevation AoA since the MU is always in the $[10^\circ - 50^\circ]$ angle sight of the AP. We know that UPA's estimation performance deteriorates as we move towards the edges. In practice, due to the density of the APs, the MUs are expected to be in the azimuth angle range of $[20^\circ - 150^\circ]$, and in the elevation angle range of $[20^\circ - 50^\circ]$, i.e., AP density of fewer than 60 meters.

C. Joint sync&loc

We perform analysis for the scenario shown in Figure 1, which is regarded in [8], [9] as challenging. A car commences its journey by accelerating to reach the velocity of 14 m/s (=50 km/h). It continues moving with constant velocity and decelerates upon approaching the intersection until it completely stops (e.g., due to the red light). The same repeats between the two intersections. At the second intersection, it begins moving, then takes a turn, and continues to accelerate to 14 m/s limit until it exits the map. All the turns, as well as the acceleration coefficients, are chosen randomly. During its journey, at each joint sync&loc round k, the MU exchanges time-stamps with a fixed number of APs (N_{AP}) in $\mathcal{I}_i$, the link to each of which is LoS/NLoS with the probability of 0.8/0.2. The APs are grouped into $\mathcal{I}_i$ based on the distance criteria, that is, $\mathcal{I}_i$ includes the N_{AP} closest APs to the i-th MU. A further assumption is that, at each joint sync&loc period T, $N_{\text{ant}} \times N_{\text{ant}}$ CIRs are available at each AP connected to the MU. In our simulations, the CIRs are obtained using the QuaDRiGa channel model. More explicitly, at each round k, knowing the true MU-AP distance and the link condition, i.e., LoS or NLoS, the CIRs are generated using the "3GPP_38.901_UMi" scenario of the QuaDRiGa channel model. Moreover, the RMSEs obtained by [8], [9] serve as the baseline to our approach. The second scheme with which we compare our proposed algorithm is the L-BRF filtering proposed in [14], [15]. The aforementioned approaches are the most relevant as they draw on the same inputs as our proposed method does.

We initialize all the clock offsets from the $\mathcal{U}(-10^3, 10^3)$ ns. The initial skews of all the clocks are drawn from the uniform distribution $\mathcal{U}(1-10^{-4}, 1+10^{-4})$, which corresponds to skew values between 0 and 100 *part-per-million* (ppm). The covariance of the clock process noise $\mathbf{Q}_n(\tilde{\boldsymbol{\vartheta}}_i)$ is set to $\text{diag}(10^{-5}, 100)$ to account for the residual errors from the previous iterations as well as the external noises on the clock skew and offset. The covariance of position process noise $\mathbf{Q}_n(\mathbf{p}_i)$ amounts to $\text{diag}((14T)^2, (14T)^2)$ to account for every possible movement of the MU. All the additional simulation parameters can be found in Table II.

Figure 10 shows the RMSE of clock offset estimation for three joint snyc&loc algorithms. The DePF algorithm is compared with two linear Bayesian methods, i.e., EKF and L-BRF, in multiple scenarios. In particular, we compute the RMSEs in three scenarios, with the number of LoS APs ranging from 1 to 3. In an additional scenario, we consider the MU being connected to three APs, where each MU-AP link condition is set to LoS with the probability of 0.8. As can be seen, for all the LoS scenarios, the L-BRF and DePF deliver an identical performance, which is expected as they rely on the same approach to estimate the clock parameters.

TABLE II
SIMULATION PARAMETERS

General parameters	Values
# of independent simulations	1000
Initial random delays ($\tilde{\theta}_i$)	$\mathcal{U}(-10^3, 10^3)$ ns
Initial random skew (γ_i)	$\mathcal{U}(1-10^{-4}, 1+10^{-4})$ or [0–100] ppm
Max. MU velocity	14 (m/s)
AP density	50 m
Distance traversed by the MU	600 m
QuaDRiGa parameters	
Scenario	3GPP_38.901_UMi
Center Frequency / FFT size (N_s)	3.8 GHz / 64
# of MU/AP antenna (N_{ant})	1 / 3 × 3
Filter Parameters	
Period of joint sync&loc (T)	100 ms
Process noise covariance matrix ($\mathbf{Q}_n$)	diag(10^{-5}, 10, 1.5, 1.5)
# of Gaussian mixtures (gdfs)	500
DNN parameters	
l_H, n_H	2, 50
Optimizer	Adam (lr= 0.001, beta_1=0.9, beta_2=0.999)
# of epochs	10
Batch size	16
Activation function of hidden layers	ReLU
Activation function of output layer	Softmax

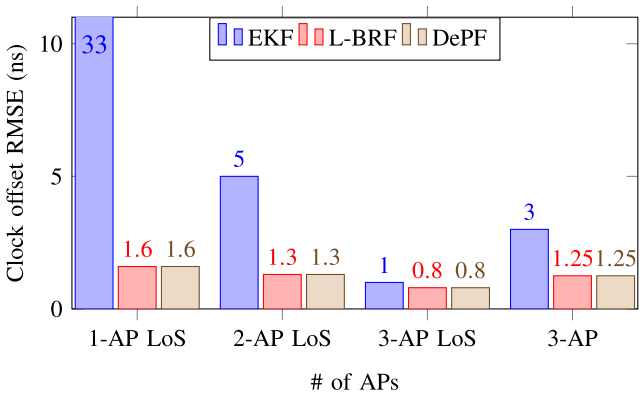


Fig. 10. Performance comparison of three joint synchronization and localization algorithms in terms of clock offset estimation.

On the other hand, the performance of the EKF falls behind as it does not explicitly draw on the synchronization signals to estimate the clock offset. Moreover, the synchronization algorithm scheme utilized to synchronize the APs, i.e., hybrid BP-BRF network synchronization, leads to a more precise inter-AP synchronization and, consequently, it lowers the MU clock offset estimation error. In the last case, the L-BRF and DePF that draw on DNN-based NLoS identification outperform the EKF-based method where the NLoS links are identified by means of Rice factor of the incoming signal strength.

Figure 11 depicts the RMSE of position estimation for three joint sync&loc algorithms. The DePF algorithm is compared with two linear Bayesian methods, i.e., EKF and L-BRF, in the same scenarios as in Figure 10. As can be seen, for almost all the scenarios, the DePF algorithm delivers superior performance. In particular, since the DePF employs

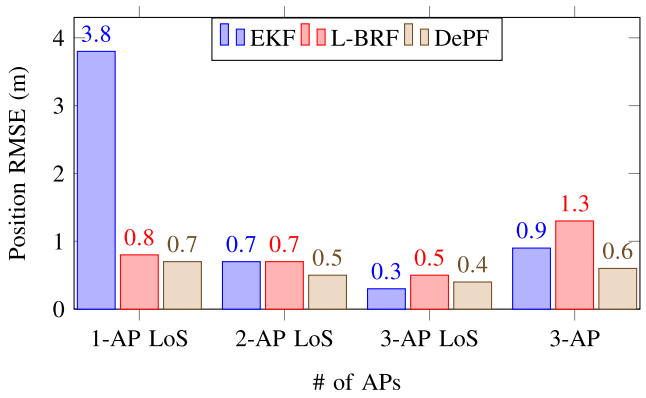


Fig. 11. Performance comparison of three joint synchronization and localization algorithms in terms of position estimation.

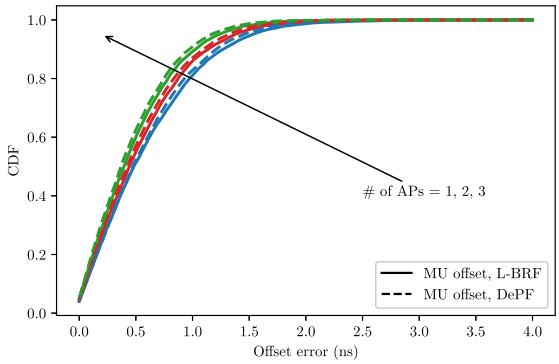


Fig. 12. Performance comparison of L-BRF and DePF when estimating the MUs' clock offset.

a higher number of gdfs, rather than only one, to approximate the posterior distribution it can estimate the position more accurately. Furthermore, DePF stands out when dealing with NLoS links. This is straightforward to notice as the RMSE of position estimation is lower for DePF in the 3-AP scenario where the L-BRF employs the same NLoS identifier as DePF. Additionally, unlike EKF and L-BRF, DePF does not need any initialization, which is of crucial importance in practice as initialization would require the APs to request position estimation from the MUs, which may not be always possible. Overall, considering 2-AP LoS, 3-AP LoS, and 3-AP scenarios, EKF and L-BRF perform close to DePF when both a reliable initialization and MU-AP links with known LoS conditions are available. Nevertheless, such assumptions are questionable in practice, rendering the EKF-based and L-BRF algorithms futile in real-world scenarios.

Hereafter, all the simulations have been carried out assuming that there is always at least one LoS MU-AP link. Figure 12 presents the CDF of the clock offset estimation error when the MU is connected to multiple APs. It can be seen that the estimation accuracy always remains below 2 ns and increases as both L-BRF and DePF utilize more measurements to estimate the clock offset and skew. In fact, since the APs are synchronized with high precision, collecting time-stamps from each additional AP does provide additional information about the statistics of MU's clock parameters and, therefore,

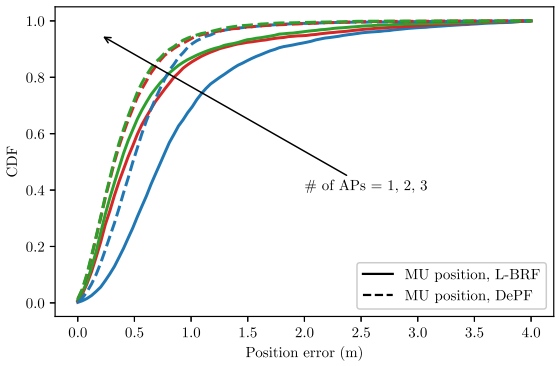


Fig. 13. Performance comparison of L-BRF and DePF when estimating the MUs' position.

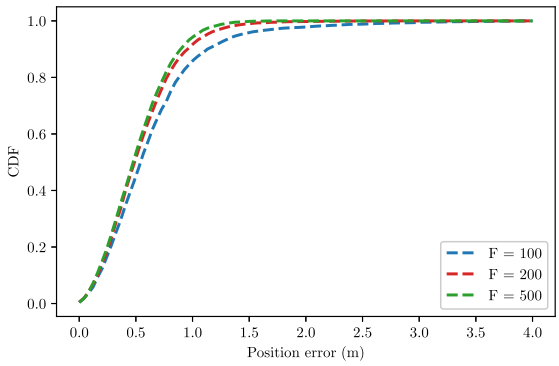


Fig. 14. Performance of joint sync&loc algorithm for different number of gdfs.

increases the accuracy of the estimation. Such precision is necessary if the location of the MU is to be accurately estimated. We note that each single ns inaccuracy maps to 0.3 m distance measurement error and, consequently, worsens the location estimation. Furthermore, the performance of both schemes is identical as they draw on the same approach, i.e., modeling the clock parameter with a single gdf, to estimate the clock parameters.

Figure 13 presents the CDF of the position estimation error when the MU is connected to multiple APs. As can be seen, the position estimation error is less than 1 meter in 90% of the cases for the DePF algorithm. We observe that DePF significantly outperforms the L-BRF, especially for 2- and 3-AP scenarios. In particular, unlike the L-BRF that approximates the posterior with a single Gaussian distribution, in DePF, the approximation is based on multiple gdfs. Consequently, the approximated posterior is closer to the true one, leading to a more precise position estimation. Another subtle observation is that, although the position estimation error decays with the growth in the number of APs, increasing the number of APs from 2 to 3 only slightly improves the performance. In fact, the third AP is normally far away from the MU, leading to a poorer (AoA and time-stamp) measurement accuracy compared to that of the first two APs. Hence, it does not provide substantial further information about the posterior distribution of the MU's location.

Figure 14 indicates the CDF of the position estimation for multiple numbers of gdfs. It can be noticed that the position estimation ameliorates with the increase of the number of gdfs. This is expected as in PGM filters the posterior distribution is approximated by multiple gdfs. Consequently, the more gdfs we employ, the more accuracy we achieve, albeit with higher computation time. Nevertheless, the error reduction is decreasing when increasing the number of gdfs, suggesting that a proper balance needs to be struck between the number of gdfs and the localization accuracy. In the scenarios presented in this work, one can achieve satisfactory performance even with 500 gdfs.

Figure 15 shows the CDF of the clock offset estimation error carried out by a single AP for different time-stamp uncertainties, i.e., $\sigma_T = 2, 4, 6$. As can be seen, the clock

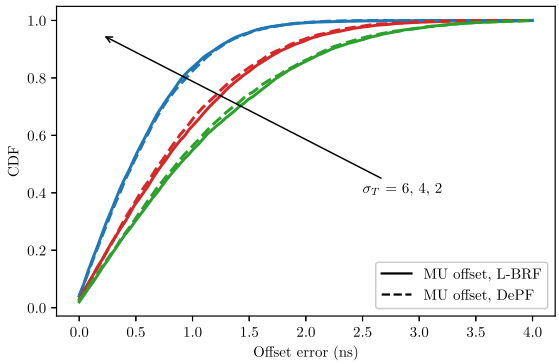


Fig. 15. Clock offset estimation performance of joint sync&loc algorithm with different number of APs involved.

offset estimation accuracy drops as the σ_T grows. It remains, however, less than 3 ns in 90% of the cases. Such degradation can cause an additional error in position estimation as, given (9), both parameters are intertwined. Specifically, the offset estimation error can introduce distance measurement error, resulting in imprecision when estimating the position. Nevertheless, the uncertainty of the time-stamping of the state-of-the-art devices is expected to be below 5 ns. Moreover, the destructive impact of the uncertainty can be also mitigated by employing more synchronized APs as discussed previously and shown in Figure 12.

Figure 16 shows the CDF of position estimation conducted by a single AP for different time-stamp accuracies. It can be noticed that the position estimation accuracy deteriorates with the growth in the time-stamp uncertainty. Specifically, the growth in uncertainty results in more erroneous distance measurements and offset estimations, which, consequently, worsens the position estimation accuracy. Nevertheless, it can be readily seen that DePF is more successful in mitigating the destructive effect of the time-stamp uncertainty. Moreover, for both DePF and L-BRF, employing more APs can alleviate the negative impact of large time-stamp uncertainty. In both Figures 15 and 16, it can be noticed that σ_T plays a decisive role in the outcome of the estimation algorithm, which also reveals the importance of hardware components in the design of a robust and precise joint

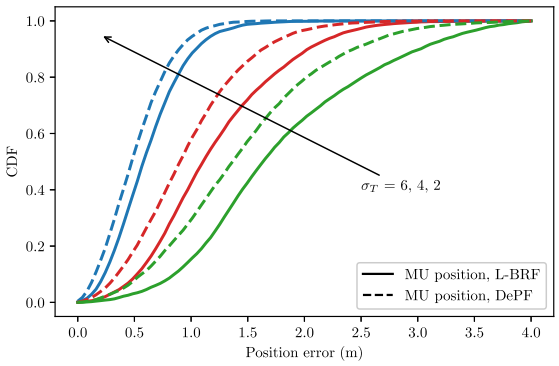


Fig. 16. Position estimation performance of joint sync&loc algorithm with different time-stamp accuracy.

sync&loc algorithm. In practice, such uncertainty in commercial off-the-shelf devices is expected to be below 5 ns.

In summary, one can see that DNNs can play a decisive role by facilitating accurate decision-making in simple, albeit crucial, tasks such as NLoS identification. Furthermore, it can be noticed that in the case when we have multiple LoS links available, the performance of the EKF-based and L-BRF approaches both in terms of clock offset and position is close to that of DePF. Nevertheless, in the absence of LoS condition, DePF demonstrates more competence in estimating the clock and position parameters by employing only a few hundred gdfs. Another point worth mentioning is that time-stamp exchange is of high potential to be employed for performing joint sync&loc. In particular, the current communication devices are capable of performing FTM up to 5 ns accuracy, fertilizing the ground for precise offset and distance measurements, which are the basis for precise joint synchronization and localization.

V. Conclusion and Future Works

We presented a DNN-assisted Particle-based filtering (DePF) algorithm for joint synchronization and localization (sync&loc) of Mobile Users (MUs) in communication networks. In particular, we leveraged an asymmetric time-stamp exchange mechanism, traditionally utilized for time synchronization, to estimate the clock offset and skew while simultaneously obtaining information about the distance between the access points and the MUs. Further on, we combined the aforementioned measurements with the angle of arrival estimation and the link condition, i.e., line-of-sight or non-line-of-sight, returned by a pretrained DNN to localize the MUs. Simulation results indicate that while the performance of the proposed algorithm is promising, especially under challenging real-world conditions, the position and clock offset estimation errors are dependent on the accuracy of hardware time-stamping. We mitigated the negative impact of this dependency by deploying more access points for performing joint sync&loc.

In this work, we drew on simulations to prove the efficiency of our proposed algorithm. However, to cross-validate the obtained results, the algorithm needs to be implemented in practice. Therefore, in future works, we will employ the hardware at our disposal to evaluate the performance of our proposed joint sync&loc algorithm in practice.

Abbreviations

AoA: Angle of Arrival, **AP:** Access Point, **BN:** Bayesian Network, **BP:** Belief Propagation, **BS:** Base Station, **CDF:** Cumulative Distribution Function, **CIR:** Channel Impulse Response, **CFR:** Channel Frequency Response, **DePF:** DNN-assisted Particle-based Bayesian Filtering, **DNN:** Deep Neural Network, **EKF:** Extended Kalman Filter, **FFT:** Fast Fourier Transform, **FTM:** fine time measurement, **gdf:** Gaussian density function, **L-BRF:** Linearized Bayesian Recursive Filtering, **LoS:** Line-of-Sight, **ML:** Machine Learning, **MU:** Mobile User, **NLoS:** Non-Line-of-Sight, **PGM:** Particle Gaussian Mixture, **RMSE:** Root Mean Square Error, **sync&loc:** Synchronization and Localization, **SVM:** Support Vector Machine, **UPA:** Uniform Linear Array.

References

[1] Y.-C. Wu, Q. Chaudhari, and E. Serpedin, "Clock synchronization of wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 28, no. 1, pp. 124–138, Dec. 2010.

[2] N. Maletic, V. Sark, J. Gutiérrez, and E. Grass, "Device localization using mmWave ranging with sub-6-assisted angle of arrival estimation," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast. (BMSB)*, Jun. 2018, pp. 1–6.

[3] *IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks-Specific Requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard 802.11-2016 (Revision IEEE Std 802.11-2012), 2016, pp. 1–3534.

[4] J. Zheng and Y.-C. Wu, "Joint time synchronization and localization of an unknown node in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1309–1320, Mar. 2009.

[5] W. Yuan, N. Wu, B. Etzlinger, H. Wang, and J. Kuang, "Cooperative joint localization and clock synchronization based on Gaussian message passing in asynchronous wireless networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 9, pp. 7258–7273, Sep. 2016.

[6] B. Etzlinger, F. Meyer, F. Hlawatsch, A. Springer, and H. Wymeersch, "Cooperative simultaneous localization and synchronization in mobile agent networks," *IEEE Trans. Signal Process.*, vol. 65, no. 14, pp. 3587–3602, Jul. 2017.

[7] F. Meyer, B. Etzlinger, Z. Liu, F. Hlawatsch, and M. Z. Win, "A scalable algorithm for network localization and synchronization," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4714–4727, Dec. 2018.

[8] J. Werner, M. Costa, A. Hakkarainen, K. Leppanen, and M. Valkama, "Joint user node positioning and clock offset estimation in 5G ultra-dense networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2015, pp. 1–7.

[9] M. Koivisto *et al.*, "Joint device positioning and clock synchronization in 5G ultra-dense networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 5, pp. 2866–2881, May 2017.

[10] S. P. Chepuri, R. T. Rajan, G. Leus, and A.-J. van der Veen, "Joint clock synchronization and ranging: Asymmetrical time-stamping and passive listening," *IEEE Signal Process. Lett.*, vol. 20, no. 1, pp. 51–54, Jan. 2012.

[11] R. M. Vaghefi and R. M. Buehrer, "Cooperative joint synchronization and localization in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 63, no. 14, pp. 3615–3627, Jul. 2015.

[12] M. Goodarzi, D. Cvetkovski, N. Maletic, J. Gutiérrez, and E. Grass, "A hybrid Bayesian approach towards clock offset and skew estimation in 5G networks," in *Proc. IEEE 31st Annu. Int. Symp. Pers., Indoor Mobile Radio Commun.*, Aug. 2020, pp. 1–7.

[13] M. Goodarzi, D. Cvetkovski, N. Maletic, J. Gutierrez, and E. Grass, "Synchronization in 5G: A Bayesian approach," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Jun. 2020, pp. 194–199.

[14] M. Goodarzi, N. Maletic, J. Gutiérrez, and E. Grass, "Bayesian joint synchronization and localization based on asymmetric time-stamp exchange," in *Proc. Int. Symp. Netw., Comput. Commun. (ISNCC)*, Oct. 2020, pp. 1–7.

[15] M. Goodarzi, D. Cvetkovski, N. Maletic, J. Gutiérrez, and E. Grass, "Synchronization in 5G networks: A hybrid Bayesian approach toward clock offset/skew estimation and its impact on localization," *EURASIP J. Wireless Commun. Netw.*, vol. 2021, no. 1, pp. 1–22, Dec. 2021.

[16] A. S. Stordal, H. A. Karlsen, G. Nævdal, H. J. Skaug, and B. Vallès, "Bridging the ensemble Kalman filter and particle filters: The adaptive Gaussian mixture filter," *Comput. Geosci.*, vol. 15, no. 2, pp. 293–305, Mar. 2011.

[17] D. L. Alspach and H. W. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximations," *IEEE Trans. Autom. Control*, vol. AC-17, no. 4, pp. 439–448, Aug. 1972.

[18] F. Gustafsson, "Particle filter theory and practice with positioning applications," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 25, no. 7, pp. 53–82, Jul. 2010.

[19] A. Doucet, N. de Freitas, K. Murphy, and S. Russell, "Rao-blackwellised particle filtering for dynamic Bayesian networks," 2013, *arXiv:1301.3853*.

[20] I. Guvenc and C.-C. Chong, "A survey on TOA based wireless localization and NLOS mitigation techniques," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 3, pp. 107–124, Aug. 2009.

[21] Y. Qi, H. Kobayashi, and H. Suda, "On time-of-arrival positioning in a multipath environment," *IEEE Trans. Veh. Technol.*, vol. 55, no. 5, pp. 1516–1526, Sep. 2006.

[22] S. Li, M. Hedley, I. B. Collings, and D. Humphrey, "Joint trajectory and ranging offset estimation for accurate tracking in NLOS environments," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 1, pp. 3–14, Feb. 2019.

[23] F. Yin, C. Fritsche, F. Gustafsson, and A. M. Zoubir, "EM- and JMAP-ML based joint estimation algorithms for robust wireless geolocation in mixed LOS/NLOS environments," *IEEE Trans. Signal Process.*, vol. 62, no. 1, pp. 168–182, Jan. 2013.

[24] B. Mondal *et al.*, "3D channel model in 3GPP," *IEEE Commun. Mag.*, vol. 53, no. 3, pp. 16–23, Mar. 2015.

[25] K. Yu and E. Dutkiewicz, "NLOS identification and mitigation for mobile tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 49, no. 3, pp. 1438–1452, Jul. 2013.

[26] S. Venkatraman and J. Caffery, "Statistical approach to non-line-of-sight BS identification," in *Proc. 5th Int. Symp. Wireless Pers. Multimedia Commun.*, vol. 1, 2002, pp. 296–300.

[27] S. Marano et al., "NLOS identification and mitigation for localization based on UWB experimental data," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 7, pp. 1026–1035, Sep. 2010.

[28] F. Xiao, Z. Guo, H. Zhu, X. Xie, and R. Wang, "AmpN: Real-time LOS/NLOS identification with WiFi," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–7.

[29] J. Heaton, *Introduction to Neural Networks With Java*. Chesterfield, MI, USA: Heaton Research, 2008.

[30] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, Apr. 2017.

[31] (2021). *AI and ML—Enablers for Beyond 5G Networks*. [Online]. Available: http://doi.org/10.5281/zenodo.4299895

[32] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propag.*, vol. AP-34, no. 3, pp. 276–280, Mar. 1986.

[33] X. F. Zhang, L. Y. Xu, L. Xu, and D. Z. Xu, "Direction of departure (DOD) and direction of arrival (DOA) estimation in MIMO radar with reduced-dimension MUSIC," *IEEE Commun. Lett.*, vol. 14, no. 12, pp. 1161–1163, Dec. 2010.

[34] R. Roy and T. Kailath, "Esprit-estimation of signal parameters via rotational invariance techniques," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 7, pp. 984–995, Jul. 1989.

[35] O. A. Oumar, M. F. Siyau, and T. P. Sattar, "Comparison between MUSIC and ESPRIT direction of arrival estimation algorithms for wireless communication systems," in *Proc. 1st Int. Conf. Future Gener. Commun. Technol.*, Dec. 2012, pp. 99–103.

[36] J. Eidson and K. Lee, "IEEE 1588 standard for a precision clock synchronization protocol for networked measurement and control systems," in *Proc. 2nd ISA/IEEE Sensors Ind. Conf.*, 2002, pp. 98–105.

[37] B. Etzlinger, H. Wymeersch, and A. Springer, "Cooperative synchronization in wireless networks," *IEEE Trans. Signal Process.*, vol. 62, no. 11, pp. 2837–2849, Jun. 2014.

[38] M. Leng and Y.-C. Wu, "Distributed clock synchronization for wireless sensor networks using belief propagation," *IEEE Trans. Signal Process.*, vol. 59, no. 11, pp. 5404–5414, Nov. 2011.

[39] G. Giorgi and C. Narduzzi, "Performance analysis of Kalman-filter-based clock synchronization in IEEE 1588 networks," *IEEE Trans. Instrum. Meas.*, vol. 60, no. 8, pp. 2902–2909, Aug. 2011.

[40] J. Du and Y.-C. Wu, "Distributed clock skew and offset estimation in wireless sensor networks: Asynchronous algorithm and convergence analysis," *IEEE Trans. Wireless Commun.*, vol. 12, no. 11, pp. 5908–5917, Nov. 2013.

[41] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[42] J. Chen, S. Guan, Y. Tong, and L. Yan, "Two-dimensional direction of arrival estimation for improved Archimedean spiral array with MUSIC algorithm," *IEEE Access*, vol. 6, pp. 49740–49745, 2018.

[43] F.-G. Yan, Z.-K. Chen, M.-J. Sun, Y. Shen, and M. Jin, "Two-dimensional direction-of-arrivals estimation based on one-dimensional search using rank deficiency principle," *Int. J. Antennas Propag.*, vol. 2015, pp. 1–8, Dec. 2015.

[44] M. Mohanna, M. L. Rabeh, E. M. Zieur, and S. Hekala, "Optimization of MUSIC algorithm for angle of arrival estimation in wireless communications," *NRIAG J. Astron. Geophys.*, vol. 2, no. 1, pp. 116–124, Jun. 2013.

[45] V. Mihajlovic and M. Petkovic, "Dynamic Bayesian networks: A state of the art," Univ. Twente Document Repository, 2001.

[46] A. L. Barker, D. E. Brown, and W. N. Martin, "Bayesian estimation and the Kalman filter," *Comput. Math. with Appl.*, vol. 30, no. 10, pp. 55–77, 1995.

[47] R. Khan, S. U. Khan, S. Khan, and M. U. A. Khan, "Localization performance evaluation of extended Kalman filter in wireless sensors network," *Proc. Comput. Sci.*, vol. 32, pp. 117–124, Jun. 2014.

[48] R. Labbe. (2019). *Kalman and Bayesian Filters in Python, 2014*. [Online]. Available: https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python

[49] R. Karlsson, T. Schön, and F. Gustafsson, "Complexity analysis of the marginalized particle filter," *IEEE Trans. Singnal Process.*, vol. 53, no. 11, pp. 4408–4411, Nov. 2005.

[50] S. Jaeckel, L. Raschkowski, K. Börner, and L. Thiele, "QuaDRiGa: A 3-D multi-cell channel model with time evolution for enabling virtual field trials," *IEEE Trans. Antennas Propag.*, vol. 62, no. 6, pp. 3242–3256, Jun. 2014.

[51] J. Shao, K. Hu, C. Wang, X. Xue, and B. Raj, "Is normalization indispensable for training deep neural network?" in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 13434–13444.

Meysam Goodarzi received the B.Sc. degree in electrical communication engineering from the K. N. Toosi University of Technology, Iran, in 2013, the M.Sc. degree in communications and multimedia engineering from the Friedrich-Alexander University of Erlangen-Nuremberg, Germany, in 2017, and the M.Sc. degree in economics from the University of Paris 1 Panthéon-Sorbonne, France, in 2022. He is currently pursuing the Ph.D. degree with the Humboldt University of Berlin, Germany. Later that year, he joined IHP GmbH—Leibniz-Institute for Innovative Microelectronics, Germany. Furthermore, he has been actively contributing to the European Horizon 2020 projects (funded by European Commission), such as 5G-XHaul, 5G-PICTURE, and 5G-CLARITY. His current research interests include resource allocation, signal processing, applications of machine learning and probabilistic graphical models in wireless communications (specifically synchronization and localization), and network economics.

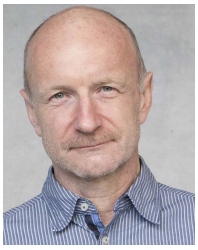
Vladica Sark received the Dipl.-Ing. degree from Ss. Cyril and Methodius University, Macedonia, in 2004, and the Dr.-Ing. degree from the Humboldt University of Berlin in 2017. Since 2004, he has been a Teaching and Research Assistant at the Faculty of Electrical Engineering, Ss. Cyril and Methodius University, where his research was focused mainly on embedded systems and smart sensors. Since 2012, he has been a Researcher with the Humboldt University of Berlin. Since 2014, he has been a Researcher at IHP—Leibniz Institute for Innovative Microelectronics, Frankfurt (Oder). His research is focused on sub-6 GHz wireless communications, precise radio localization on mmWave and terahertz wireless communications, and on mmWave RADAR systems.

Nebojsa Maletic (Member, IEEE) received the Dipl.-Ing. degree in electrical engineering (major in microwave engineering) and the M.Sc. degree in electrical engineering and computer science from the University of Belgrade, Serbia, in 2008 and 2010, respectively. In 2010, he joined the Faculty of Electrical Engineering, University of Banja Luka, Bosnia-Herzegovina, as a Teaching and Research Associate. He joined IHP GmbH—Leibniz-Institute for Innovative Microelectronics, Frankfurt (Oder), Germany, in late 2015. In IHP, he is a member of the Wireless Broadband Communications Group. He participated in several European H2020 projects (5G-XHaul, 5G-PICTURE, WORTECS, 5GENESIS, and 5G-VICTORI) related to mmWaves and their applications. His current research interests include millimeter-wave (mmWave) communications, MIMO, signal processing algorithms, hardware impairments, and the design of wireless communication systems operating in mmWave and sub-THz bands. He is a member of IEEE ComSoc Society and MTT-S Society.

Jesús Gutiérrez Terán (Member, IEEE) received the B.S. and Ph.D. degrees in telecommunication engineering from the University of Cantabria in 2008 and 2013, respectively. Since 2013, he has been a Post-Doctoral Researcher with IHP GmbH—Leibniz-Institute for Innovative Microelectronics, Frankfurt (Oder), Germany. He has participated in several EU funded projects, being the project coordinator of 5G-PICTURE, 5G-VICTORI, and 5G-CLARITY. His research interests include digital signal processing for high performance hardware architectures, millimeter wave systems, wireless networks, 5G mobile communication, localization and positioning systems, and synchronization.

Giuseppe Caire (Fellow, IEEE) was born in Torino, in 1965. He received the B.Sc. degree in electrical engineering from the Politecnico di Torino in 1990, the M.Sc. degree in electrical engineering from Princeton University in 1992, and the Ph.D. degree from the Politecnico di Torino in 1994.

He was a Post-Doctoral Research Fellow with the European Space Agency (ESTEC), Noordwijk, The Netherlands, from 1994 to 1995; an Assistant Professor in telecommunications with the Politecnico di Torino; an Associate Professor at the University of Parma, Italy; a Professor at the Department of Mobile Communications, Eurecom Institute, Sophia-Antipolis, France; and a Professor of electrical engineering with the Viterbi School of Engineering, University of Southern California, Los Angeles. He is currently an Alexander von Humboldt Professor with the Faculty of Electrical Engineering and Computer Science, Technical University of Berlin, Germany. His main research interests are in the field of communications theory, information theory, and channel and source coding, with particular focus on wireless communications. He received the Jack Neubauer Best System Paper Award from the IEEE Vehicular Technology Society in 2003, the IEEE Communications Society and Information Theory Society Joint Paper Award in 2004 and 2011, the Okawa Research Award in 2006, the Alexander von Humboldt Professorship in 2014, the Vodafone Innovation Prize in 2015, an ERC Advanced Grant in 2018, the Leonard G. Abraham Prize for Best IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS Paper in 2019, the IEEE Communications Society Edwin Howard Armstrong Achievement Award in 2020, and the 2021 Leibniz Prize of the German National Science Foundation (DFG). He has served on the Board of Governors for the IEEE Information Theory Society from 2004 to 2007 and as an Officer from 2008 to 2013. He was the President of the IEEE Information Theory Society in 2011.



Eckhard Grass received the Dr.-Ing. degree in electronics from the Humboldt University of Berlin, Germany, in 1993.

He was a Visiting Research Fellow at Loughborough University, U.K., from 1993 to 1995, and a Senior Lecturer in microelectronics at the University of Westminster, London, U.K., from 1995 to 1999. Since 1999, he has been with IHP GmbH—Leibniz-Institute for Innovative Microelectronics, leading several projects on the implementation of wireless broadband communication systems. He is currently a Team Leader with the Wireless Broadband Communications Group, IHP. He is also a Professor with the Department of Computer Science, Humboldt University of Berlin. He has published more than 100 papers at international conferences and international journals. He was actively involved in the definition of the international mmWave standards IEEE802.15.3c and IEEE802.11ad. He is actively involved in the definition and development of 5G and 6G wireless communication systems and has coordinated several large European projects. His research topics include wireless communication systems, digital signal processing algorithms, architectures, and joint communications and sensing (JCAS).